

Answer on Question #60927, Programming & Computer Science

1. encrypt the information word by word.
2. use the last alphabetic character in each word as the key for encrypting that word.
3. consider the value of each alphabetic character as its alphabetic position i.e.
a=A=1
b=B=2
and so on
z=Z=26
1. for every word use the last alphabetic character in the word as the key. replace all the alphabetic character in the word with new alphabetic characters formed by subtracting the key value from each character.
2. key character (last alphabetic character in each word) should not be encrypted.
3. encryption should be performed only on alphabetic characters and not on non-alphabetic character such as digits or special character.
4. all alphabetic characters should be preserved as uppercase in their respective position but should be replaced with the new character.

Solution.

I build 2 class: Main for I/o, and Encrypting for Function.

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Encrypting e = new Encrypting();
        Scanner in = new Scanner(System.in);
        String firststring = in.nextLine();
        char [] myCharFirst = firststring.toCharArray();
        System.out.println("Final String: " + e.encrypting(firststring));
    }
}
public class Encrypting {
    String encrypting (String myStr)
    {
        String FinalStr;
        char [] myCharFirst = myStr.toCharArray();
        char [] myCharFinal = new char[myStr.length()];
        int LastNumber=0, nextNumber=0;
        for(int i=0; i<myStr.length(); i++)
        {
            if(i>myStr.length()) break;
            nextNumber=getNumberOfLastWord(myCharFirst,LastNumber);
            if(nextNumber>myStr.length()) break;

            for(int j=LastNumber; j<nextNumber;j++)
            {
                if(myCharFirst[j]<123 && myCharFirst[j]>96 && j!=nextNumber-
1)
                    {
                        if((myCharFirst[nextNumber-1]-myCharFirst[j])<=0)
                        {
                            myCharFinal[j]=(char) (97+(myCharFirst[j]-
myCharFirst[nextNumber-1]));
                        }
                        else
                        {
                            myCharFinal[j]=(char) (-myCharFirst[j]-
myCharFirst[nextNumber-1])+96);
                        }
                    }
            }
        }
    }
}
```

```

        else
        {
            myCharFinal[j]=myCharFirst[j];
        }
    }
    LastNumber=nextNumber+1;
    i=LastNumber;
}
FinalStr=new String(myCharFinal);
return FinalStr;
}
int getNumberOfLastWord(char[] myChar ,int lastNumber)
{
    int i=0;
    for (i=lastNumber; i<myChar.length;i++)
    {
        if(i>myChar.length) break;
        if(myChar[i]==32)
        {
            break;
        }
    }

    return i;
}
}
}

```

Screen:

```

asddas 111 ASD
Final String: raors 111 ASD

```